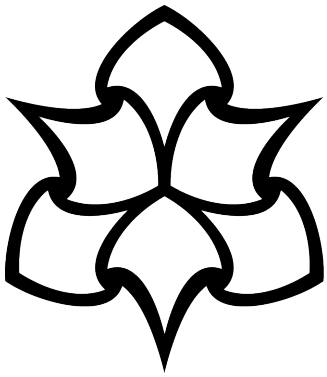

HRS-CE: A hybrid framework to integrate content embeddings in recommender systems for cold start items

Fahad Anwaar[a], Naima Iltaf[a], Hammad Afzal[a,*], Raheel Nawaz[b]

[a] National University of Sciences and Technology, Islamabad, Pakistan
[b] Manchester Metropolitan University, UK

ARTICLE INFO

Article history:
Received 3 January 2018
Received in revised form 13 August 2018
Accepted 15 September 2018
Available online 20 September 2018

Keywords:
Recommender system
Collaborative filtering
Word2vec
Cold start
User profile
Natural language processing

ABSTRACT

Recommender systems (RSs) provide the personalized recommendations to users for specific items in a wide range of applications such as e-commerce, media recommendations and social networking applications. Collaborative Filtering (CF) and Content Based (CB) Filtering are two methods which have been employed in implementing the recommender systems. CF suffers from Cold Start (CS) problem where no rating records (Complete Cold Start CSS) or very few records (Incomplete Cold Start ICS) are available for newly coming users and items. The performance of CB methods relies on good feature extraction methods so that the item descriptions can be used to measure items similarity as well as for user profiling. This paper addresses the CS problem by providing a novel way of integrating content embeddings in CF. The proposed algorithm (HRS-CE) generates the user profiles that depict the type of content in which a particular user is interested. The word embedding model (Word2Vec) is used to produce distributed representation of items descriptions. The higher representation for an item description, obtained using content embeddings, are combined with similarity techniques to perform rating predictions. The proposed method is evaluated on two public benchmark datasets (MovieLens 100k and MovieLens 20M). The results demonstrate that the proposed model outperforms the state of the art recommender system models for CS items.

1. Introduction

Recommender systems (RSs) play a significant role in e-commerce services such as recommending products e.g. books, movies, news, garments etc. Recently, they have been applied in vast area of other applications as well such as social networking, web pages and articles recommendation. Big organizations in the world such as Facebook, LinkedIn, Netflix, eBay, Amazon and many other companies build their own recommender system to obtain the preferences of their potential consumers. The business oriented success of such companies is largely dependent on the performance of their recommender systems; for instance, Netflix system recommends similar movies of interest to their users, Amazon and eBay use the recommender system to show the similar products of interest to their customers; in social networks such as Facebook, recommending the pages of interest or showing the ads of interest is the product of their recommender system that generates the major portion of their revenue [1,2].

The approaches for RSs are usually categorized as Collaborative Filtering (CF), Content-Based Filtering (CB) and the Hybrid Filtering [3–5]. CF utilizes the historical data of item recommendations from users, capturing the user's behaviour and preferences. The similarity between users on the basis of their preferences is used to recommend new items to the users without analyzing the content of items [6,7]. On other hand, the CB filtering utilizes the description of items by considering their characteristics and attributes to match user profiles. Some similarity metric is used to compare new item with the previously liked items by the user, contained in the user profile and best matches of items are recommended to the users. However, CB filtering is dependent on item's metadata (rich description of products) and structured user profiles for making recommendations to users [8,9]. Hybrid filtering exploits the semantic of the contents as well as the user preferences to take the benefits of both CF and CB approaches which consequently increases the performance of recommender systems. Many promising algorithms in above categories are reported in literature, however the complexity of system still requires improvement on certain issues.

* Corresponding author.
E-mail addresses: fahadanwaar.mscs22@students.mcs.edu.pk (F. Anwaar), naima@mcs.edu.pk (N. Iltaf), hammad.afzal@mcs.edu.pk (H. Afzal), r.nawaz@mmu.ac.uk (R. Nawaz).

https://doi.org/10.1016/j.jocs.2018.09.008
1877-7503/

One of the most challenging problems in RSs is the cold start problem. The cold start problem is related to significant degradation of recommendation quality when new users or new items come into the system. CF needs reasonable amount of rating records of user on certain items to make recommendations. However, when new user or item comes into the system, CF fails to make effective recommendations due to sparsity of information. The reasonable amount of work has been done to solve the user cold start problem by incorporating user demographic information. However, cold start item problem in recommender systems still remains an open research issue and requires much improvement to make accurate recommendations [10,11].

This paper addresses the problem of cold start items. The proposed method predicts the ratings for cold start items by incorporating implicit data (ratings) and auxiliary information (item content). The word embedding model, Word2vec (W2V) [34] and its variant for composite data are used to extract the content features of the items. The higher representation for each item description, represented as the resultant vector of Word2Vec, is used to generate the user profiles. The user profiles depict the user's taste and likings. The proposed algorithm integrates the content features of the items into the memory based collaborative filtering to predict the ratings for CS items. The achieved results demonstrate the efficient performance of proposed system. The major contributions of our work can be summarized as:

- A hybrid framework, HRS-CE is proposed that extracts item descriptions using content embedding methods i.e. Word2Vec. The content features are integrated with memory based collaborative filtering to predict ratings.
- HRS-CE utilizes the detailed descriptions instead of using only meta-data such as tags, keywords; thus capturing the deep semantics of item descriptions that result in better user profiles, and therefore, better predictions.
- The performance comparison shows that the proposed method outperforms other state of the art methods for cold start items.

The remaining paper is organized as follows: Section 2 provides the related work; proposed recommendation model is described in Section 3; performance evaluation and analysis of experimental results are presented in Section 4. Finally Section 5 concludes the paper.

2. Related work

The cold start problem is related to the sparsity of data for new items and users. A variety of matrix factorization (MF) techniques such as Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NNMF) and Tensor Factorization [12] have been applied to CF to solve the sparsity problem. MF attempts to factorize the user-item rating matrix into lower dimensional item and user latent vectors. The Stochastic Gradient Descent (SGD) [5] and Alternating Least Squares (ALS) [13] are the optimization algorithms used as a learning algorithms in MF. In [14], the probabilistic matrix factorization (PMF) is proposed which outperforms the SVD model and also provides linear scalability on big datasets. Recently different variations and generalization based on PMF are also proposed such as generalized PMF [15] and Bayesian PMF [16]. There are other methods that incorporate the "web of trust" for users. For example, in [12], authors proposed a novel framework called "Merge" which incorporates only trusted neighbors into traditional collaborative filtering approaches. The ratings of trusted neighbors are merged to aggregate and represent the preferences of active users. In [17], a trust based MF (TrustSVD) technique is proposed which extends the SVD++ by exploiting both implicit and explicit impact of trusted users on the prediction of items for an active user. However, these models are unable to solve cold start problem effectively.

The content based (CB) methods [18] address the cold start problem for new items as they utilize the contents (item descriptions) as well; thus new items are handled easily. However, it does not address the problem with new users effectively. Recently, hybrid models that combine the CF with the user or item content are proposed for CS problem. In [19], latent factor model (TopicMF) based on biased matrix factorization model is proposed. TopicMF combines the latent factors in rating information along with topics in user-review text to handle the data sparsity in better manner as textual review consists of richer information as compared to alone rating information. In [20], functional matrix factorization (FMF) approach is proposed for cold start problem. This approach learns the user profiles from the interview process. Latent Dirichlet application (LDA) is applied to learn the content features of an item. However, FMF works only on implicit rating prediction problem and cannot learn latent representation successfully under high sparsity of content information. A similar work is reported in [21], where a Feature Based Regression Algorithm (FRBE) is proposed that includes side information (user gender and age) of all user and item to deal with cold start problem. However, the features of users do not depict the user interest effectively.

The recently proposed CB methods attempt to develop user profiles from other channels such as tagging systems [22] and social trust network [23]. In [24], the latent factor model is proposed which incorporates user and item metadata into modified matrix factorization. The comprehensive framework performs well on warm and cold start users, however, its performance degrades for cold start items. In [26], an interest sequence based collaborative filtering (ISCF) recommendation, built on users' interest sequences (IS) is proposed. It ranks the user's ratings and online behaviors with respect to the timestamps. It captures the user interest sequences which are more dynamic in nature, and therefore, provides better accuracy.

In [27], a User Rating Profile model (URP) is proposed which is based on a procreative latent variable model for rating based collaborative filtering. Their latent variable model views the rating-based data at the level of user rating profiles. The URP model predicts the ratings for the items that a user has not rated, based on other users rating profiles. However, their model fails to predict rating for cold start item when there is no rating available for newly coming item in any of the available user rating profiles. A profile based framework for learning path discovery is proposed in [28] which assists the group of learners to acquire new knowledge. Their proposed group model reflects the characteristics and values of group learners which helps to identify a suitable learning path in an e-learning environment. However, the proposed framework requires a sufficient user information otherwise proposed model leads to a data sparsity problem. A multi level user profiling method is proposed in [29] which integrates both, the user ratings and tags information to get the personalized search. A three level User Profiling (TUP) method is based on user's favorite, ordinary and annoying tags which helps to alleviate the current limitations in collaborative tagging systems for personalized search.

Recently, the Cross Domain Recommender Systems (CDRS) are introduced which assist the recommendations in a target domain based on knowledge learned from a source domain. In CDRS, the items in the source domain are recommended to users of the target domain. To cope with the cold start problem, cross domain methods collect the auxiliary information from other domains and alleviate the cold start user-item problem effectively [30]. A joint rating and popularity prediction framework for cold start item is proposed in [31]. The joint prediction framework exploits the sentinel users' reviews on the cold start item to elicit their latent profiles. The

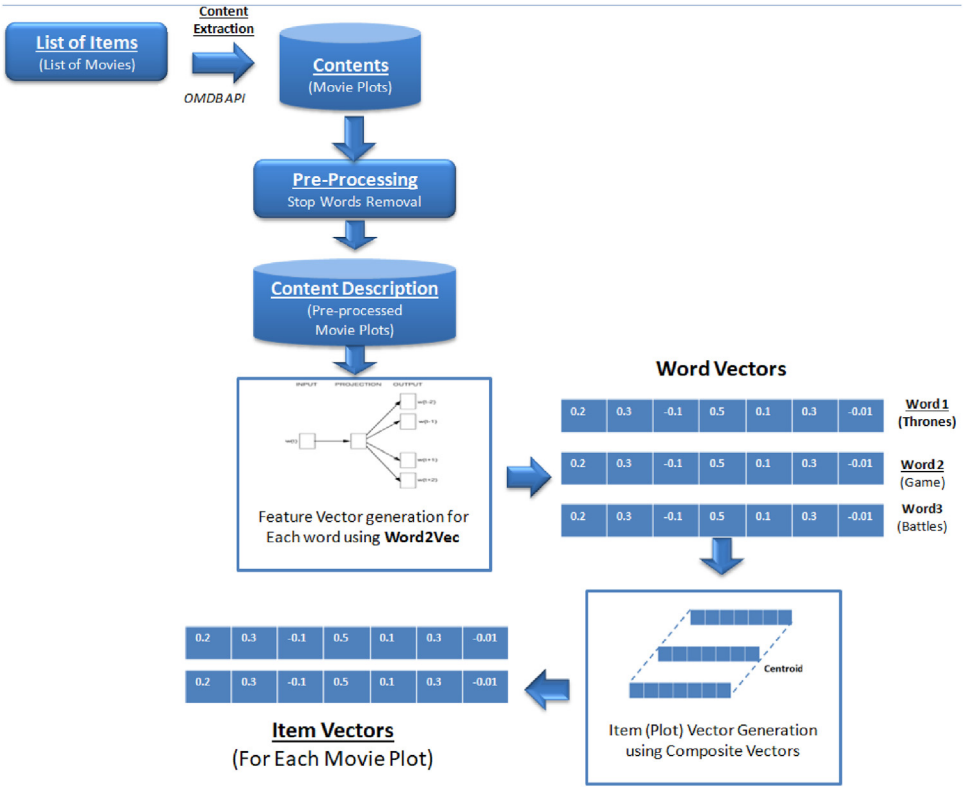

Fig. 1. Creation of item feature vectors using content based embedding (Word2Vec Model [34]).

extracted latent profiles from user reviews are used to simultaneously predict user-specific ratings and popularity of the cold start items. However, their framework performance declines for joint prediction as the content features of cold start item is not used.

Recently, deep learning based methods have achieved greater success in domains such as image processing, computer vision, and artificial intelligence. Such methods are now attracting researchers attention in recommender systems as well. In [32], a collaborative deep learning model (CDL) is proposed that incorporates the Stack Denoising Auto Encoder (SDAE) into CF based latent factor model for recommendation. However, CDL utilizes a very simple CF model which gives only top-N recommendations. In [33], another collaborative filtering and deep learning based framework for complete cold start and incomplete cold start is proposed. It extracts the content features of item by utilizing deep neural network. SDAE and CF model (named as timeSVD++) is modified to take content features to predict the ratings. The proposed method performs well under high sparsity of user ratings but it utilizes Weighted Bag-of-Word model. The traditional methods such as Bag-of-Words, however, are unable to capture semantic similarity of words.

3. Hybrid framework for rating prediction using content embeddings

A hybrid recommender system using content embeddings, named as HRS-CE, is proposed to predict the ratings for cold start items. The proposed system is built upon word embedding based content extraction for item descriptions, which are then used to build the user profiles. This section provides the system description in textual as well as mathematical form. The overall methodology is illustrated in Figs. 1 and 2 . In order to showcase the efficiency of the system, the experiments are performed on MovieLens dataset; where each movie is considered as an item and movie plot is considered as the content description. An example is provided to elaborate the functionality of HRS-CE for movie recommendation.

3.1. System description

The HRS-CE utilizes the item descriptions (auxiliary information) to obtain raw content information of all items. The auxiliary information, e.g. plots of movies, is required for each item. The obtained information is processed using pre-processing techniques including text cleaning using stopwords removal. Stopwords refer to commonly used words (such as "*the*") in language that give minor useful information. The processed information is then used to produce the distributed representations of item descriptions as vector notations based on word embedding technique *Word2vec* [34]. Word2Vec model is trained with Hierarchical Softmax. We initialized the model with following parameters, $min\_count = 1$, $window = 5$, and $size = 300$ where size is N dimensional feature vector, min_count value is the minimum frequency of words which are considered in context. The process of creation of item vectors is shown in Fig. 1 where the highlighted labels depict the name of processes/items while the text inside brackets show the particular instance from the example movie recommendation system.

The feature vectors of items and mean ratings of items given by existing users are used to generate the user profiles. The neighborhood is computed for newly coming item (x) from user profiles using cosine similarity measure. Rating for item (x) is predicted using collaborative filtering technique. The overall system architecture is presented in Fig. 2.

3.2. Mathematical formulation of problem

Let $U = [u_1, u_2, u_3, \ldots, u_m]$ be the set of all users, $I = [\hat{i}_1, \hat{i}_2, \hat{i}_3, \ldots, \hat{i}_n]$ be the set of all non-CS items and let $X = [x_1, x_2, x_3, \ldots, x_f]$ be the set of all CS items. The two dimensional user-item rating matrix is represented as

$$R_{u\hat{i}} \in \{1, 2, 3, 4, 5\}^{m \times n},$$

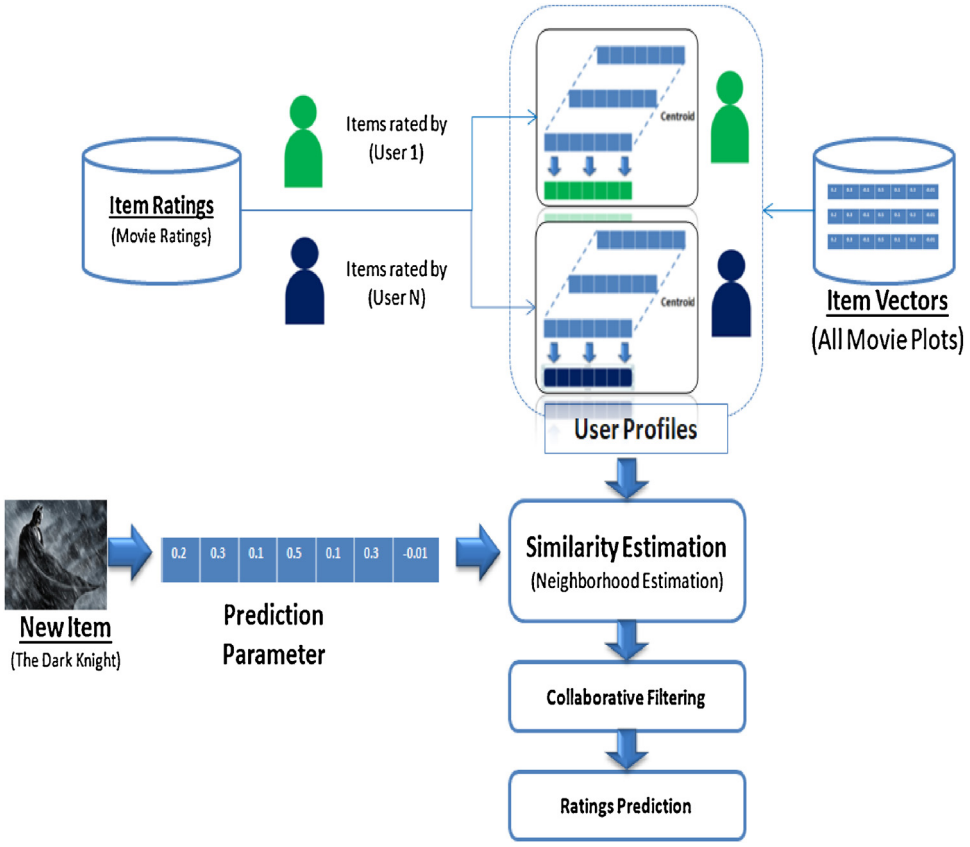

Fig. 2. System flow of proposed system.

where $R_{u\hat{i}}$ represents the rating given by user $u \in U$ on item $\hat{i} \in I$, m is the number of users and n is the number of non-CS items.

The proposed recommender task is to predict unknown ratings $\hat{R}_{ux}$ for (x) item based on known rating $R_{u\hat{i}}$. Let G be the item description (i.e. plot of a movie) which is composed of words such as $w_1, w_2, w_3, \ldots, w_l$.

$$G \underset{\text{Plot}}{\longleftarrow} \underset{\text{Description}}{\{w_1, w_2, w_3, \ldots, w_l\}}$$

From textual description of plots, a Corpus C is built, comprising of all movies (G) and is represented as:

$$C \underset{\text{Corpus}}{\longleftarrow} \underset{\text{Generation}}{\{G_1, G_2, G_3, \ldots, G_n\}}$$

where $C \in \{w_1, w_2, w_3, \ldots, w_l, w_{l+1}, \ldots, w_t\}$. t represents the total number of words in the whole corpus C.

The proposed framework maximizes the log probability under a sequence of training words such as $w_1, w_2, w_3, \ldots, w_t$ for a given corpus C.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t) \tag{1}$$

where $T = |C|$, the size of Corpus; c is the size of training window.

The hierarchical Softmax [34] is used to define the basic probability $p(w_{t+j}|w_t)$ of the output word:

$$p(w_{t+j}|w_t) = \Pi_{i=1}^{L(w_{t+j})-1}\{\sigma(1(n(w_{t+j}, i+1) = child(n(w_{t+j}, i))).v_{n(w_{t+j},i)}.v_{w_t})\} \tag{2}$$

where

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

- $L(w)$ represents the length of the path from root to word w in the binary tree so that $n(w, 1)$ = root and $n(w, L(w)) = w$.
- *child* (n) is the fixed child of node n.
- v_n is the vector representation of inner node.
- v_{w_t} is the input vector of word w_t.
- The identity function is represented by 1 which is 1 if x is true else it is -1.

Using Eqs. (1) and (2), the vector representation of each word w, represented as (V_w), is created. Since the averaging of word embedding is most suited to obtain the composite representation of text (such as sentences and paragraphs) and their similarity [34], the proposed technique computes the higher vector notation for each item $(\hat{i})$ as:

$$V_{G_{\hat{i}}} = \frac{1}{|l|}\sum_{w=1}^{l} V_l \tag{3}$$

where $\hat{i} \in [\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_n]$. l is set of words describing the properties of an item and $l \neq 0$.

The proposed model utilizes the content feature of an item $V_{G_{\hat{i}}}$ and known rating matrix $R_{u\hat{i}}$ to make rating prediction on unknown item (x). The mean rating $(\bar{R}_{u\hat{i}})$ and auxiliary information $(\bar{V}_{G_{\hat{i}}})$ is combined to generate user profile (P_U) for each user:

$$P_U \underset{\text{user profile}}{\longleftarrow} \{\bar{V}_{G_{\hat{i}}}, \bar{R}_{u\hat{i}}\} \tag{4}$$

For user profile generation, only a subset of items are considered for which user has given ratings equal to threshold (θ). θ is tuned to depict the user preference $\theta \in [1-5]$. $\bar{V}_{G_{\hat{i}}}$ is computed by tak-

Table 1
Example: user's ratings for movies.

Users/movies	Deep impact	Babe	Dante's peak	Red planet
Alice = α	3	3	4	3.5
Bob = β	2.5	5	3	5
John = γ	5	2	3.5	2.5

ing centroid of vector representation of items having rating greater than threshold:

$$\bar{V}_{G_{\hat{i}}} = \frac{1}{|A|}\sum_{j=1}^{A} V'_{G_j} \tag{5}$$

where $V'_{G_j} \subset V_{G_{\hat{i}}}, j \subset \hat{i}$ and A is total number of items present in users profile.

To predict the rating for CS item (x), the proposed algorithm uses adjusted cosine similarity measure to relate CS item (x) to non CS items $(\hat{i})$. Based on item feature vector obtained from Eq. (5), the adjusted cosine similarity formula is used to compute the similarity between non CS items $(\hat{i})$ and CS item (x). For any two feature vectors $\bar{V}_{G_{\hat{i}}}$ and $\bar{V}_{G_x}$ of items $(\hat{i})$ and (x), the adjusted cosine similarity is computed as:

$$S_{(\hat{i},x)} = \frac{\sum_{z=1}^{d}(\bar{V}_{G_{\hat{i}z}} - \hat{V}_{G_{\hat{i}}}).(\bar{V}_{Gxz} - \hat{V}_{G_x})}{\sqrt{\sum_{z=1}^{d}(\bar{V}_{G_{\hat{i}z}} - \hat{V}_{G_{\hat{i}}})^2}\sqrt{\sum_{z=1}^{d}(\bar{V}_{Gxz} - \hat{V}_{G_x})^2}} \tag{6}$$

where $\hat{V}_{G_{\hat{i}}}$ and $\hat{V}_{G_x}$ represents the mean values of $\bar{V}_{G_{\hat{i}}}$ and $\bar{V}_{G_x}$ vectors.

Let $M^S(u, x)$ represents the M most similar non CS items $(\hat{i})$ among the all $(\hat{i})$ items to a CS item (x). The final prediction of rating for (x) item is computed:

$$\hat{R}_{ux} = \frac{\sum_{\hat{i}\in M^S(u,x)} R_{u\hat{i}}.S_{(\hat{i},x)}}{\sum_{\hat{i}\in M^S(u,x)} S_{(\hat{i},x)}} \tag{7}$$

where $R_{u\hat{i}}$ represents actual ratings available in training set.

3.3. An illustrative example of working of HRS-CE

The working of proposed framework, HRS-CE, is demonstrated using an illustrative example comprising a scenario with four movies (Deep Impact, Babe, Dante's Peak and Red Planet) rated by three users (Alice, Bob and John) as shown in Table 1. The values of ratings range from 1 to 5. The plots of the four movies, comprising short textual descriptions on OMDB, are extracted using OMDB API. The plots are pre-processed to remove stop-words and resultant descriptions are represented as G1, G2, G3 and G4, corresponding to the four movies respectively. These descriptions are then used to extract content embeddings using word2vec, thus generating the vectorial representation as V_1, V_2, V_3 and V_4 for the four movies respectively. The process is shown in Table 2.

The user profile for each user P_α, P_β and P_γ is calculated using the vectorial representations of only those movies which are rated equal or greater than 3.5 by the corresponding user as shown in Table 3.

Table 3
Example: user profiles using movies rated equal/greater 3.5.

User profiles	Mean of vectors embeddings; average ratings
P_α	$\hat{Pi}$ = [(V3 + V4)/2, 3.75]
P_β	$\hat{Pi}$ = [(V2 + V4)/2, 5]
P_γ	$\hat{Pi}$ = [(V1 + V3)/2, 4.25]

Table 4
Plot description and embedding of a non-rated movie (cold start item).

User profiles	Mean of vectors embeddings; average ratings
P_α	P = [(V3 + V4)/2, 3.75]
P_β	P = [(V2 + V4)/2, 5]
P_γ	P = [(V1 + V3)/2, 4.25]

Now, for a new incoming movie, e.g. Armageddon (a cold start item), which is not rated by the users (Alice, Bob or John), HRS-CE predicts its rating for all users as follows: First, the feature vector, i.e. V5 of Armageddon is computed using the extracted plot from OMDB API as shown in Table 4. Using the user profiles P_α, P_β and P_γ and feature vector V5, the neighborhood is determined based on maximum score of cosine similarity (Eq. (6)) between V5 and user profiles as Sim(P_α,V5), Sim(P_β,V5) and Sim(P_γ,V5). After getting top similarity measure for V5 with existing user profiles, collaborative filtering technique is applied to predict the rating for new movie against each existing user using Eq. (7).

3.4. Computational complexity analysis

The training complexity to compute the vectorial representation for whole Corpus C comprising the textual descriptions of the plots, using word2vec can be computed as $W = E * T * (F * (D + D * log_2 V)$ where E is number of the training epochs, T is the number of the words in the Corpus, V is size of vocabulary, F and D are the parameters used in calculating word2vec that represent the size of window and projection layer respectively. This complexity is computed for skip-gram model using the calculations provided in [34]. Given that W is the complexity of extracting content embeddings; for total I items and M users, the complexity of creating all user profiles, represented as U is $W * I * M$. This is the training phase which is performed only once offline. For a newly coming cold start item, the complexity to predict its rating by performing collaborative filtering is $O(U * I)$.

4. Performance evaluation

This section presents the performance evaluation of the proposed HRS-CE system. The description of the datasets is presented, followed by the steps taken for preparation of the datasets for experiments. The experimental settings and results are discussed, followed by comparison with the state of the art methods.

Table 2
Plot Description and Embeddings of Rated Movies

Movies	Plot description (pre-processed)	Vector embeddings
Deep Impact	G1 = comet destroyed colliding Earth allowed shelters survive people survive	V1 = Word2Vec{G1}
Babe	G2 = Babe pig raised sheepdogs learns herd sheep little help Farmer Hoggett.	V2 = Word2Vec{G2}
Dante's Peak	G3 = vulcanologist arrives countryside town named second desirable place live America discovers long dormant volcano Dante's Peak wakeup moment.	V3 = Word2Vec{G3}
Red Planet	G4 = Astronauts robotic dog AMEE Autonomous Mapping Evaluation Evasion search solutions save dying Earth searching Mars mission terribly awry.	V4 = Word2Vec{G4}

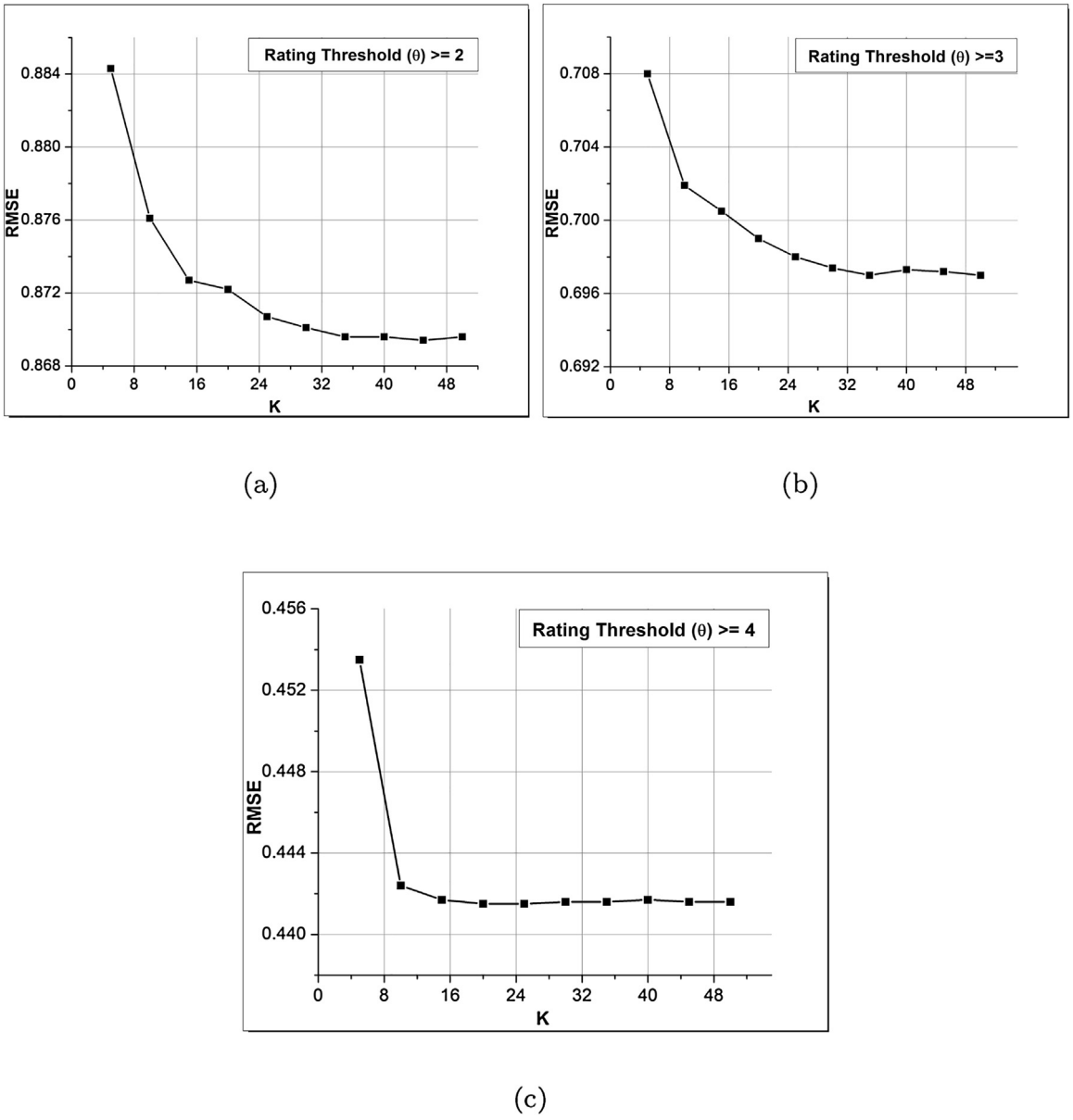

Fig. 3. Performance of proposed model on 100k with varying neighbours K and rating threshold θ.

4.1. Dataset preparation

Two datasets, *MovieLens* (100k) & (20M), have been most popularly used for the evaluation of RSs. [30] showed that these datasets have been used in around 22% of the studies in the field of RSs. Therefore, we have selected these two datasets to showcase the performance of HRS-CE. The sparsity level of both datasets vary between ≈93.7% to ≈99.46%. The 20M dataset is recommended for new research work by *MovieLens* [35] and is extremely sparse about ≈99.46% which makes it well suited for cold start item problem. The *Movielens* (20M) dataset consists of 20 million explicit ratings $R_{u\hat{i}} \in \{0.5, 1, 2, 3, 4, 5\}$ and 465,564 tag applications made by 138,493 users to 27,278 movies. The *MovieLens* (100k) dataset contains 100K ratings $R_{u\hat{i}} \in \{1, 2, 3, 4, 5\}$, given by 943 users on 1682 movies. We have used only those target users in the experiment who have rated at least 20 movies in both 100k and 20M dataset. The statistical information about datasets is shown in Table 5.

The auxiliary information for movie plots is required for each item to predict the rating for CS items; however, the original datasets do not contain the plots of movies. Therefore, in order to perform experiments, the original datasets are extended to include movie plots which are retrieved using OMDB API.[1] To extract movie plots from OMDB database, python based script is written which traverses the movies in the *MovieLens* datasets and corresponding plots are extracted by automatically sending search request against each movie title and year to OMDB database. The extracted movie plots are filtered by removing stopwords. The plots are then converted into vector space representation using Word2vec. In our

Table 5
The statistics of the MovieLens datasets.

Datasets	Users	Items	Ratings	Rating scale	Density
20M	138,493	27,278	20,000,263	[0.5–5]	0.54%
100k	943	1,682	100,000	[1–5]	6.30%

[1] http://www.omdbapi.com

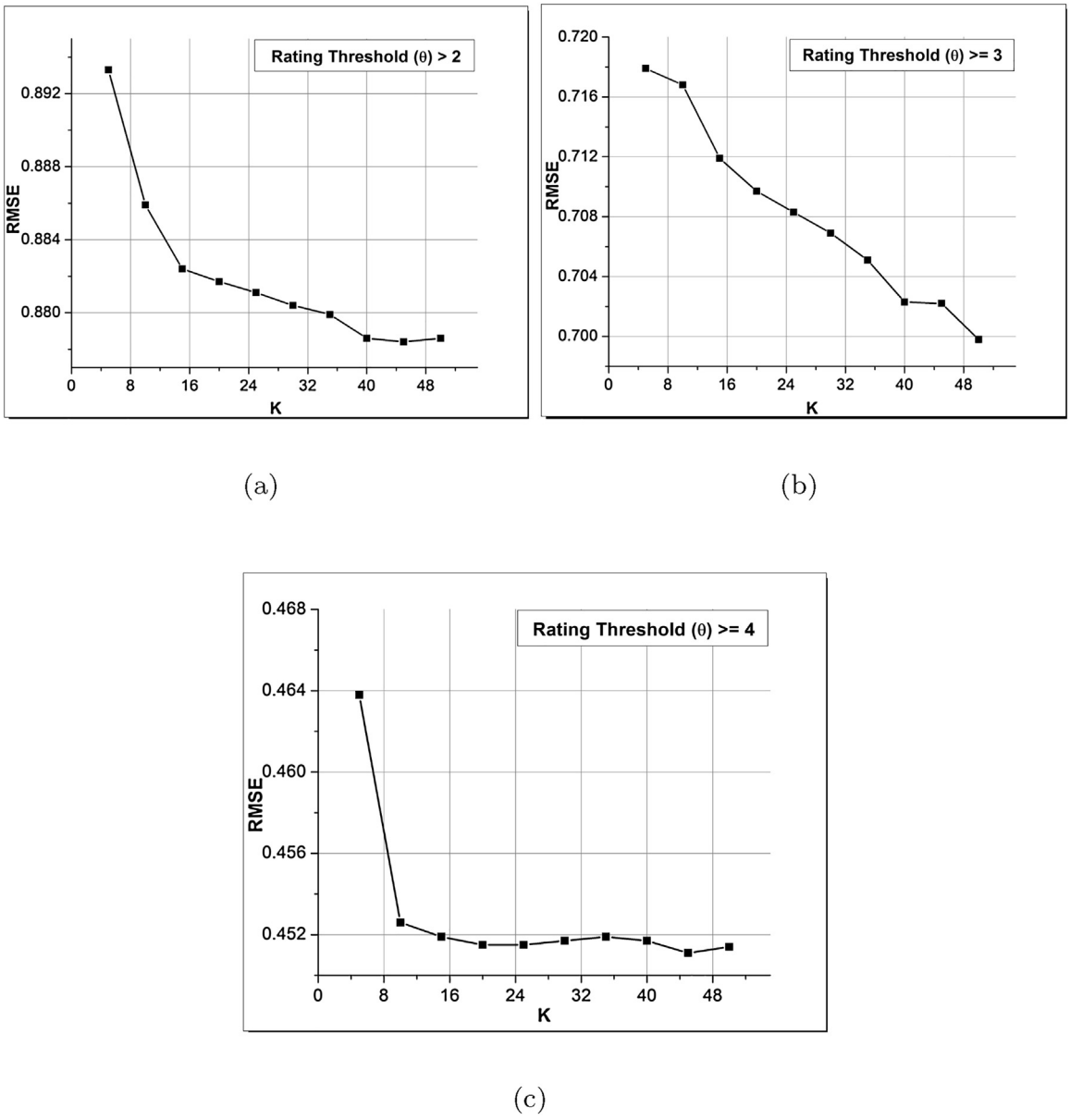

Fig. 4. Performance of proposed model on 20M with varying neighbours K and rating threshold θ.

experiments, we have used all the words as missing any word may result in the loss of information. Plot vector $V_{G_{\hat{i}}}$ is computed from each word vector V_l using Eq. (3). Those movies for which the plot is not available in OMDB database are removed along with their ratings and users from datasets.

To simulate the cold start item scenario (existing users/newly coming items), we divide the items into two disjoint subsets, training and test set. The datasets are divided into Training and Test sets using Holdout method by keeping 80% of items for training and the remaining items for testing. The training set is used only to determine the representative users. Then the interest level of existing users on new items is predicted for test set items. In other words, the training set is used to calculate prediction using proposed algorithm while test set is used to assess the prediction performance of proposed algorithm.

4.2. Evaluation metric

Different evaluation metrics are traditionally used to measure the performance of recommender systems. In our experiments, a widely used metric, i.e. Root Mean Square Error (RMSE) is used. RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,x}(R_{ux} - \hat{R}_{ux})^2} \tag{8}$$

where N represents the total number of predicted ratings, $\hat{R}_{ux}$ represents predicted rating and $R_{u\hat{x}}$ represents known rating on item i given by user u. The lower RMSE shows better recommendation accuracy.

In addition to RMSE, we have also used precision and recall metrics, commonly used in information retrieval, to measure the recommendation quality.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

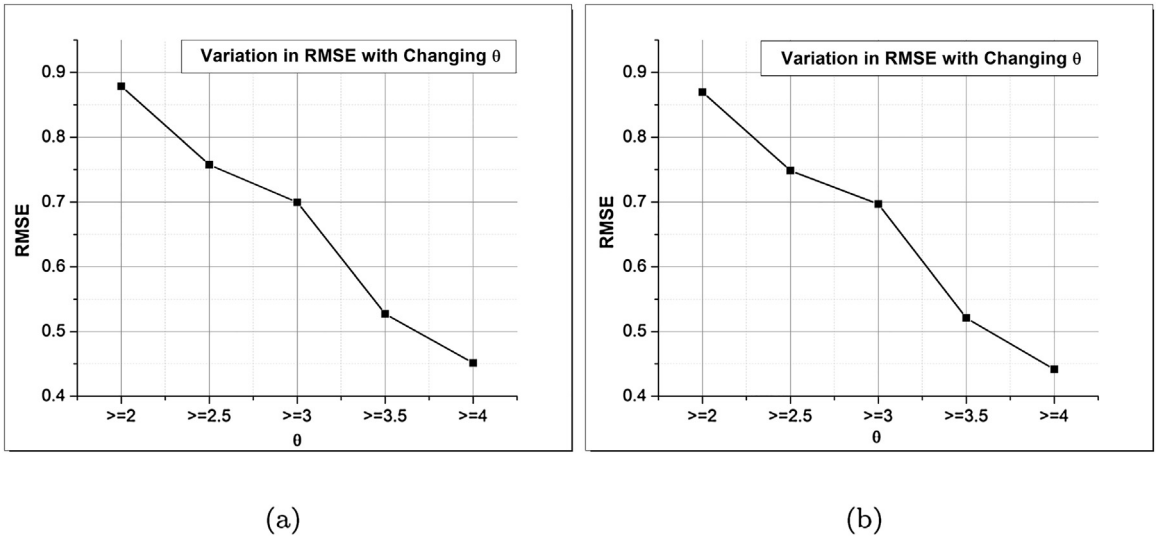

Fig. 5. Performance of Proposed Model with varying the rating threshold to create user profiles: (a) Dataset 20M, (b) Dataset 100k.

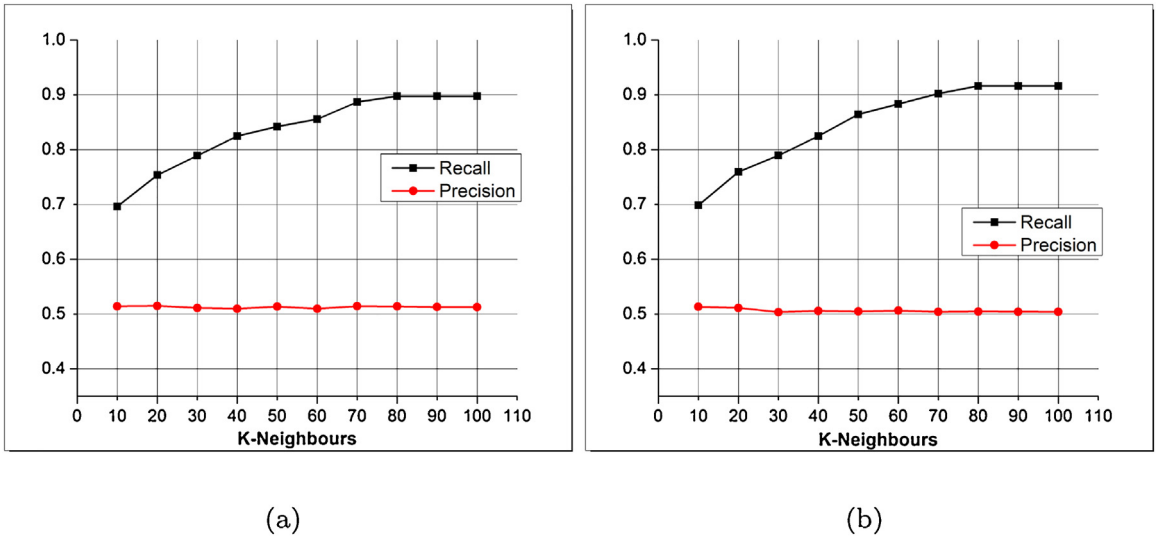

Fig. 6. Precision and Recall with varying the K neighbours: (a) Dataset 20M, (b) Dataset 100k.

where, *TP* denotes true positive (item relevant and recommended). *FP* is false positive (item irrelevant and recommended). *FN* is false negative (item relevant and not recommended).

To differentiate the relevant and irrelevant items, we mark the items with rating above 3 as relevant and those rated below 3 out of 5 as irrelevant to the user.

4.3. Results and analysis

The performance of proposed technique is evaluated on *Movielens*100*k* and *Movielens*20*M* datasets using 5-fold cross validation. A series of experiments are performed to study the impact of number of K nearest neighbors. K refers to the most similar users that are used to perform rating prediction, as shown in Eq. (6). In both sets of experiments (Figs. 3 and 4), the value of K is varied from 5 to 50. It is evident from Figs. 3 and 4 that the value of RMSE decreases as the value of K is increased; thus considering more number of similar neighbors while performing predictions improves the results.

For user profile generation, only a subset of items (θ) are considered that depict the user preference. Fig. 5 demonstrate the effect of varying the value of θ on both datasets, keeping the value of K at 50. The best values of RMSE are obtained at high value of θ, i.e. $\theta = 4$ and most number of neighbours, i.e. $K = 50$;

In order to compute the precision and recall, the algorithm first discretize the ratings by converting them into two classes; i.e. liked and disliked. An item with a rating greater than 3 is defined as liked, whereas, a rating less or equal to 3 is defined as disliked. The proposed framework HRS-CE also yields better results in terms of top-N recommendations as shown in Fig. 6. The precision and recall values given for top-N recommendation at different values of K neighbors are consistently higher in proposed framework.

The performance of proposed model is also compared with other best performing hybrid models, reported in literature so far (to the best of our knowledge), designed for cold start problem in recommender system. Table 6 summarizes the comparison of proposed model with existing hybrid models including [24], [36]. Gogna et al. technique [24] incorporated the user and item metadata into modified matrix factorization. Strub et al. [36] provided a collaborative filtering based neural network model which computes the non linear matrix factorization from side information and sparse input ratings. Their technique, V-CFN++ reported a result ≈0.7652 RMSE using 20M dataset. It is evident from Table 6 that proposed model

Table 6
Performance comparison of proposed model with other best performing systems for cold start items.

Model	Dataset	RMSE
Cheng et al. [26]	100k	0.9510
Xu et al. [25]	1M	0.9453
Mnih et al. [14]	20M	0.9373
Gogna et al. [24]	100k	0.9214
Koren et al. [38]	20M	0.8721
Nguyen et al. [37]	20M	0.8528
Strub et al. [36]	20M	0.7652
HRS-CE	100k	**0.5217**
HRS-CE	20M	**0.5272**

provides largely improved result using 20M dataset for cold start item problem. Table 6 also compares the prediction performance of proposed system with state-of-the-art methods, reported in literature, for CS items including CF based, CB based and Hybrid models. The model proposed by Cheng et al. [26] is built on user interest sequences and ranked user ratings and behaviors with respect to timestamps. It has reported a result of ≈0.9510 RMSE on 100k dataset. The Xu et al. technique [25] provides novel rating comparison strategy which exploits knowledge from warm items or users to calibrate the latent profiles of cold start items or users, reported a result of 0.9453 RMSE on 1M dataset. The Mnih et al. [14] technique provides CF based rating prediction model which reported a result of 0.9373 RMSE on 20M dataset. The Koren et al. [38] provides SVD++ model which utilizes both explicit and implicit feedback in rating prediction and reported a result of 0.8721 RMSE on 20M dataset (much better than earlier approaches). The Nguyen et al. technique [37] combines explicit and implicit feedback in unified model for rating prediction and gives 0.8528 RMSE on 20M dataset. However, the proposed model gives 0.5219 and 0.5271 RMSE on 100k and 20M dataset respectively under cold start item condition which gives largely improved results as compared to above described state of the art approaches.

The reason for good performance of HRS-CE is that other described best performing systems for cold start items in literature do not incorporate the rich auxiliary information to alleviate cold start item problem. For example, Gogna et al. technique [24] incorporates the movie genres (Action, Thriller, Animation, Mystery, etc.) as an item metadata, Strub et al. [36] incorporates the genres and tags as an item metadata. The Nguyen et al. technique [37] incorporates the implicit feedback (e.g., clicks, views, movie rental history) and explicit feedback like rating profiles in the unified model for rating prediction. This unified model combines the matrix factorization framework and item embedding to alleviate the cold start item problem. In [14,25], explicit feedback such as rating profiles whereas, in [38], both explicit and implicit feedback of an item are incorporated in the matrix factorization based collaborative filtering framework to predict the ratings for non-rated items. However, the proposed HRS-CE utilizes the detailed descriptions of movie plot instead of using only item meta-data such as tags, keywords, genres, views, clicks, likes, movie rental history; thus capturing the deep semantics of item descriptions that result in better user profiles, and therefore, better predictions.

5. Conclusion

In this paper, a content embedding based hybrid recommender model is proposed. The model predicts the ratings for cold start items by exploiting items textual description. Word embedding model is used to produce distributed representation of items description. After computing content similarity between feature vectors, most related items are utilized for rating prediction of cold start items using memory based collaborative filtering technique. Our model is evaluated on MovieLens dataset with plot descriptions, however, it is applicable to other types of data where item descriptions are available, e.g. online shopping such as Amazon and eBay, the product descriptions can be utilized to create content features. The proposed model is compared with the state of art techniques [14,24–26,36–38]. Experimental results depict that the proposed model (HRS-CE) improves the recommendation performance for CS items and outperforms the state of art techniques. The use of detailed item descriptions, along with content embedding, captures the deep semantics of items as well, thus provides much better results than using only meta-data based descriptions. The proposed recommender framework is not only limited to movie domain, it can also be applicable to other domains such as books, music, web pages, newspapers, research article, sports, tv shows, documents and tourism scenic spots based recommendations. However, the rich auxiliary information about an item is a prerequisite for proposed algorithm to predict the rating for cold start items. In this study we only focus on cold start item problem, in future we intend to extend our model to cold start user and system cold start problems.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[2] P.G. Campos, F. Díez, I. Cantador, Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols, User Model. User-Adapt. Interact. 24 (1–2) (2005) 67–119.
[3] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender Systems Handbook, Springer, New York, NY, USA, 2011, pp. 1–35.
[4] Y. Koren, R. Bell, Advances in collaborative filtering, in: Recommender Systems Handbook, Springer, New York, NY, USA, 2011, pp. 145–186.
[5] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009).
[6] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, Proceedings of the 10th International Conference on World Wide Web (2001) 285–295.
[7] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. 2009 (2009) 4.
[8] J. Wang, A.P. de Vries, M.J.T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (2006) 501–508.
[9] M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation, Commun. ACM 40 (3) (1997) 66–72.
[10] J. Bennett, C. Elkan, B. Liu, P. Smyth, D. Tikk, KDD cup and workshop 2007, ACM SIGKDD Expl. Newslett. 9 (2) (2007) 51–52.
[11] D. Zhang, C.H. Hsu, M. Chen, Q. Chen, N. Xiong, J. Lloret, Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems, IEEE Trans. Emerg. Top. Comput. 2 (2) (2014) 239–250.
[12] G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowl.-Based Syst. 57 (2014) 57–68.
[13] Y. Zhou, D. Wilkinson, R. Schreiber, R. Pan, Large-scale parallel collaborative filtering for the Netflix prize Lecture Notes in Computer Science, vol. 5034, 2008, pp. 337–348.
[14] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Adv. Neural Information Processing Systems, 2008, pp. 1257–1264.
[15] H. Shan, A. Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, 2010 IEEE 10th International Conference on Data Mining (ICDM) (2010) 1025–1030.
[16] R. Salakhutdinov, A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, Proceedings of the 25th International Conference on Machine Learning (2008) 880–887.
[17] J. Zhang, G. Guo, N. Yorke-Smith, TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, Aaai (2015) 123–129.
[18] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, 2011, pp. 73–105.
[19] Y. Bao, H. Fang, J. Zhang, TopicMF: simultaneously exploiting ratings and reviews for recommendation, AAAI, vol. 14 (2014) 2–8.
[20] K. Zhou, S.H. Yang, H. Zha, Functional matrix factorizations for cold-start recommendation, Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (2011) 315–324.
[21] X. Xu, Z. Lizhong, Z. Xiaowei, X. Zhenzhen, L. Yu, A feature-based regression algorithm for cold-start recommendation, J. Ind. Prod. Eng. 31 (1) (2014) 17–26.

[22] Z.K. Zhang, C. Liu, Y.C. Zhang, T. Zhou, Solving the cold-start problem in recommender systems with social tags, EPL (Europhys. Lett.) 92 (2) (2010) 28002.
[23] H. Ma, I. King, M.R. Lyu, Learning to recommend with explicit and implicit social relations, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 29.
[24] A. Gogna, A. Majumdar, A comprehensive recommender system model: improving accuracy for both warm and cold start users, IEEE Access 3 (2015) 2803–2813.
[25] J. Xu, Y. Yao, H. Tong, X. Tao, J. Lu, Ice-breaking: mitigating cold-start recommendation problem by rating comparison, IJCAI (2015) 3981–3987.
[26] W. Cheng, G. Yin, Y. Dong, H. Dong, W. Zhang, Collaborative filtering recommendation on users' interest sequences, PLOS ONE 11 (5) (2016) e0155739.
[27] B.M. Marlin, Modeling user rating profiles for collaborative filtering, in: Advances in Neural Information Processing Systems, 2004, pp. 627–634.
[28] H. Xie, D. Zou, F.L. Wang, T.L. Wong, Y. Rao, S.H. Wang, Discover learning path for group users: a profile-based approach, Neurocomputing 254 (2017) 59–70.
[29] Q. Du, H. Xie, Y. Cai, H.F. Leung, Q. Li, H. Min, F.L. Wang, Folksonomy-based personalized search by hybrid user profiles in multiple levels, Neurocomputing 204 (2016) 142–152.
[30] M.M. Khan, R. Ibrahim, I. Ghani, Cross domain recommender systems: a systematic literature review, ACM Comput. Surv. (CSUR) 50 (3) (2017) 36.
[31] Z. Miao, J. Yan, K. Chen, X. Yang, H. Zha, W. Zhang, Joint prediction of rating and popularity for cold-start item by sentinel user selection, IEEE Access 4 (2016) 8500–8513.
[32] H. Wang, N. Wang, D.Y. Yeung, Collaborative deep learning for recommender systems, Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015) 1235–1244.
[33] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. 69 (2017) 29–39.
[34] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[35] F.M. Harper, J.A. Konstan, The MovieLens datasets: history and context, ACM Trans. Interact. Intell. Syst. (TiiS) 5 (4) (2016) 19.
[36] F. Strub, J. Mary, R. Gaudel, Hybrid Collaborative Filtering with Autoencoders, 2016. arXiv preprint. arXiv:1603.00806.
[37] T. Nguyen, K. Aihara, A. Takasu, A Probabilistic Model for Collaborative Filtering with Implicit and Explicit Feedback Data, 2017. arXiv preprint. arXiv:1705.02085.
[38] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2008) 426–434.

Fahad Anwaar has completed his bachelor in Computer Software Engineering and currently is doing his Master's degree in Computer Software Engineering under the supervision of Dr. Naima Iltaf and Dr. Hammad Afzal. His field of interest includes Machine learning, Recommender System, Data Science and Android Application development.

Dr. Naima Iltaf is an Assistant Professor at National University of Sciences and Technology, Pakistan. She received her PhD in Software Engineering from National University of Sciences and Technology, Pakistan in 2013. Her field of interest encompasses Recommender System, Trust Management, Image Processing.

Dr. Hammad Afzal is currently heading "The Center of Data and Text Engineering and Mining" (CoDTeEM) group at NUST. His primary interests are machine learning, text and data mining systems. He completed PhD from School of Computer Science, University of Manchester, UK in Dec, 2009 under supervision of Dr. Goran Nenadic in Text Mining Group. Before PhD, he completed MSc in Advanced Computing Sciences from University of Manchester, UK where he was awarded Program Prize of the year from Department of Computation for acquiring highest grades in MSc courses. He has also been affiliated with Digital Enterprise Research Institute (DERI), National University of Ireland, Galway as a Research Assistant from July, 2009 to Dec, 2009.

Dr. Raheel Nawaz is Senior Lecturer at School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, UK. He has previously been associated with The National Centre for Text Mining (NaCTeM) at University of Manchester, UK where he completed his PhD degree and later worked as Research Fellow.